

TRANSMITTAL OF APPEAL BRIEF (Large Entity)

Docket No.
200-0667

In Re Application Of: Joseph G. Walacavage et al.

Application No.

09/966,121

Filing Date

September 28, 2007

Examiner

K. Griebresilassie

Customer No.

33481

Group Art Unit

2128

Confirmation No.

4437

Invention: **METHOD OF EMULATING MACHINE TOOL
BEHAVIOR FOR PROGRAMMABLE LOGIC
CONTROLLER LOGICAL VERIFICATION SYSTEM**

COMMISSIONER FOR PATENTS:

Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed on:

The fee for filing this Appeal Brief is: \$500.00

- ☐ A check in the amount of the fee is enclosed.
- ☐ The Director has already been authorized to charge fees in this application to a Deposit Account.
- ☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 06-1510. I have enclosed a duplicate copy of this sheet.
- ☐ Payment by credit card. Form PTO-2038 is attached.

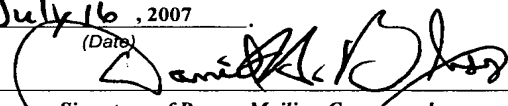
WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.


SignatureDated: July 16, 2007

Daniel H. Bliss (Reg. No. 32,398) [0693.00258]
Bliss McGlynn, P.C.
2075 West Big Beaver Road, Suite 600
Troy, Michigan 48084
(248) 649-6090

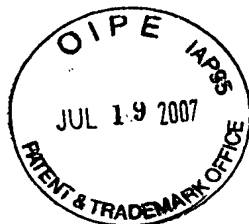
Record I.D. 81047711,

CC:

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on	
<u>July 16</u> , 2007 (Date)	
Signature of Person Mailing Correspondence	
Daniel H. Bliss	
Typed or Printed Name of Person Mailing Correspondence	

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Art Unit: 2128)
Examiner: K. Gebresilassie)
Applicant(s): J. G. Walacavage et al.)
Serial No.: 09/966,121)
Filing Date: September 28, 2001)
For: METHOD OF EMULATING)
MACHINE TOOL BEHAVIOR FOR)
PROGRAMMABLE LOGIC CONTROLLER)
LOGICAL VERIFICATION SYSTEM)



APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

By Notice of Appeal filed May 16, 2007, Applicants have appealed the Final Rejection dated February 16, 2007 and submit this brief in support of that appeal.

REAL PARTY IN INTEREST

The real party in interest is the Assignee, Ford Global Technologies, Inc.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences regarding the present application.

07/20/2007 SSITHIB1 00000028 061510 09966121
01 FC:1402 500.00 DA

CERTIFICATE OF MAILING: (37 C.F.R. 1.8) I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the U.S. Postal Service with sufficient postage as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450 on July 16, 2007, by Daniel H. Bliss

STATUS OF CLAIMS

Claims 1 through 19 have been rejected.

Claims 1 through 19 are being appealed.

STATUS OF AMENDMENTS

An Amendment Under 37 C.F.R. 1.116 was filed on April 16, 2007 in response to the Final Office Action dated February 16, 2007. An Advisory Action dated May 7, 2007 was issued and did not indicate that the Amendment under 37 C.F.R. 1.116 would be entered. A Notice of Appeal, along with the requisite fee, was filed on May 16, 2007. The Appeal Brief, along with the requisite fee, is submitted herewith.

SUMMARY OF THE CLAIMED SUBJECT MATTER**Independent Claim 1**

The claimed subject matter of independent claim 1 is directed to a method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle. [Referring to FIG. 2, a method, according to the present invention, of emulating machine tool behavior as part of the PLC logical verification system 18 is shown. In general, the method includes generating transformational arrays based on CAD geometries during the design phase of the machinery for the mechanical model. The transformational arrays are movies of manipulation of individual components in the mechanical model and are generated with the mechanical tool design system 16. These transformational arrays are then associated with the particular piece of machine component, such as a clamp,

throughout the life cycle of the design and verification process. Once the transformational arrays exist and the mechanical design is implemented on a CAD system to produce the mechanical model, the method includes exporting the mechanical model to a special purpose viewer or motion player 30 such as VisLine. Within the motion player 30, these transformational arrays are sequenced to give a first pass rendition of what the overall machine or manufacturing line behavior will be. After the user 12 is satisfied that the behavior is acceptable or the one desired, then all this information is exported to the controls system design system 17. The controls system design system 17 converts the sequence of the transformational arrays to PLC code. The user 12, using the controls system design system 17, has now substituted the sequencing logic of the transformational arrays that was present in the first pass rendition with the PLC code that is intended to be executed on the plant floor. The user 12 exports the PLC code to the PLC emulator 20 to play and visualize the PLC code. By playing this PLC code in the PLC emulator 20, and through the binding that has been preserved through this process, the user 12 can observe the sequencing of these transformational arrays using the actual PLC code as if they were watching a machine or manufacturing line of a vehicle assembly plant floor.] (FIG. 2; Specification, page 9, line 6 through page 11, line 6).

The method includes the steps of constructing a mechanical model using a computer. [The method also includes using the mechanical tool design system to construct a mechanical model 26. The mechanical model 26 is a portion of a simulated assembly or manufacturing line representing actual machines, as opposed to control hardware. The mechanical model 26 contains CAD geometries and mechanical motion from the electro-mechanical model. The geometry representation is defined through the use of (NURB) type equations. In the present invention, a user 12 uses a computer 14 for carrying out the emulation. The computer 14 sends and receives information from a mechanical tool design system 16 via an

electronic link. The mechanical tool design system 16 is a computer aided design (CAD) system and provides engineering data for standard components, tools, fixture models, and robots to create a mechanical model. The mechanical tool design system 16 allows the user 12 to define a set of kinematics or motions for the geometry of individual components in the mechanical model.] (FIGS. 1 and 2; Specification, page 7, lines 14 through 22 and page 11, line 16 through page 12, line 2).

The method also includes the steps of generating transformational arrays for the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer. [The method includes generating transformational arrays 28 for the mechanical model 26 with the mechanical tool design system 16. The transformational arrays 28 are a set of coordinates in each of a plurality of arrays. The set of coordinates comprise six coordinates of three dimensions and three rotations. The transformational arrays 28 translate or transform the original coordinates for each point on the CAD geometry into a linear file of recorded coordinates. The transformational arrays are generated by incrementally recording one position of a specific piece of geometry as it is moved through space over a period of time. It should be appreciated that the user 12 of the CAD software for the mechanical tool design system 16 is the creator of the path of the mechanical element in the mechanical model.] (FIGS. 1 and 2; Specification, page 12, lines 3 through 16).

The method further includes the steps of viewing motion of the mechanical model in a motion viewer based on the transformational arrays using the computer. [The method includes constructing a motion file based on the mechanical model 26 and the transformational arrays 28 and viewing motion of the mechanical model 26 based on the transformational arrays 28 using a motion player 30 to play the motion file. A motion file is a series of transformations

that represent the motion path of a three-dimensional object through simulation space. The motion player 30 interfaces with the mechanical model 26 to control when and in what direction a given motion file plays. The motion player 30 is a software product known as VisLine that provides a lightweight visual animation capability based on CAD three-dimensional models and sets of transformations called “motion files” derived from the CAD models. The order and time for the playing of a motion file is determined by executing a program written in the PLC emulator 20. The written program is a language specification supported by the PLC emulator 20, which is capable of communicating a complex sequence for the execution of motion files and is based on the execution of specified events according to specified conditions. It should be appreciated that events cause actions to occur and create signals available as conditions for executing other events. It should also be appreciated that the transformation does not change nature of object or part but moves the object or part in space.] (FIGS. 1 and 2; Specification, page 12, line 17 through page 13, line 18).

The method includes the steps of determining whether the motion of the mechanical model is acceptable. [The method further includes determining whether the motion of the mechanical model 26 is acceptable. The user 12 views the motion or behavior of the mechanical model 26 and determines whether the behavior is acceptable or desired.] (FIGS. 1 and 2; Specification, page 13, lines 19 through 22).

The method includes the steps of replicating the motion of the mechanical model by generating a PLC code for the motion of the mechanical model using the computer if the motion of the mechanical model was acceptable. [If the behavior is acceptable, the method exports all of this information to the controls system design system 17, which converts this information to PLC code to build a manufacturing line. The user 12 replicates the motion of the mechanical model 26 previously defined in the mechanical model 26 with motion commands in

PLC code using the controls system design system 17. If the user 12 determines that the behavior is not acceptable or desired, the method includes returning to the mechanical tool design system 16 to change the mechanical model 26 as illustrated in FIG. 3.] (FIGS. 1 through 3; Specification, page 13, line 22 through page 14, line 10).

The method includes the steps of using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator. [After the sequencing logic of the transformational arrays has been replicated with PLC code, the user 12 exports the PLC code to the PLC emulator 20 to play and visualize the PLC code. The user 12 compares the behavior of the PLC code to the accepted motion of the mechanical model 26 through the sequencing of the transformational arrays 28 as if they were watching a machine or manufacturing line of a vehicle assembly plant floor. It should be appreciated that the method is carried out on the computer 14 by the user 12.] (FIGS. 1 and 2; Specification, page 14, lines 11 through 14).

Independent claim 11

The claimed subject matter of independent claim 11 is directed to a method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle. [Referring to FIG. 2, a method, according to the present invention, of emulating machine tool behavior as part of the PLC logical verification system 18 is shown. In general, the method includes generating transformational arrays based on CAD geometries during the design phase of the machinery for the mechanical model. The transformational arrays are movies of manipulation of individual components in the mechanical model and are generated with the mechanical tool design system 16. These transformational

arrays are then associated with the particular piece of machine component, such as a clamp, throughout the life cycle of the design and verification process. Once the transformational arrays exist and the mechanical design is implemented on a CAD system to produce the mechanical model, the method includes exporting the mechanical model to a special purpose viewer or motion player 30 such as VisLine. Within the motion player 30, these transformational arrays are sequenced to give a first pass rendition of what the overall machine or manufacturing line behavior will be. After the user 12 is satisfied that the behavior is acceptable or the one desired, then all this information is exported to the controls system design system 17. The controls system design system 17 converts the sequence of the transformational arrays to PLC code. The user 12, using the controls system design system 17, has now substituted the sequencing logic of the transformational arrays that was present in the first pass rendition with the PLC code that is intended to be executed on the plant floor. The user 12 exports the PLC code to the PLC emulator 20 to play and visualize the PLC code. By playing this PLC code in the PLC emulator 20, and through the binding that has been preserved through this process, the user 12 can observe the sequencing of these transformational arrays using the actual PLC code as if they were watching a machine or manufacturing line of a vehicle assembly plant floor.] (FIG. 2; Specification, page 9, line 6 through page 11, line 6).

The method includes the steps of constructing a mechanical model using a computer. [The method also includes using the mechanical tool design system to construct a mechanical model 26. The mechanical model 26 is a portion of a simulated assembly or manufacturing line representing actual machines, as opposed to control hardware. The mechanical model 26 contains CAD geometries and mechanical motion from the electro-mechanical model. The geometry representation is defined through the use of (NURB) type equations. In the present invention, a user 12 uses a computer 14 for carrying out the emulation.

The computer 14 sends and receives information from a mechanical tool design system 16 via an electronic link. The mechanical tool design system 16 is a computer aided design (CAD) system and provides engineering data for standard components, tools, fixture models, and robots to create a mechanical model. The mechanical tool design system 16 allows the user 12 to define a set of kinematics or motions for the geometry of individual components in the mechanical model.] (FIGS. 1 and 2; Specification, page 7, lines 14 through 22 and page 11, line 16 through page 12, line 2).

The method also includes the steps of generating CAD transformational arrays for motion in the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer. [The method includes generating transformational arrays 28 for the mechanical model 26 with the mechanical tool design system 16. The transformational arrays 28 are a set of coordinates in each of a plurality of arrays. The set of coordinates comprise six coordinates of three dimensions and three rotations. The transformational arrays 28 translate or transform the original coordinates for each point on the CAD geometry into a linear file of recorded coordinates. The transformational arrays are generated by incrementally recording one position of a specific piece of geometry as it is moved through space over a period of time. It should be appreciated that the user 12 of the CAD software for the mechanical tool design system 16 is the creator of the path of the mechanical element in the mechanical model.] (FIG. 1; Specification, page 12, lines 3 through 16).

The method includes the steps of constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer and viewing the motion of the motion file in a motion viewer using the computer. [The method includes constructing a motion file based on the mechanical model 26 and the transformational arrays 28

and viewing motion of the mechanical model 26 based on the transformational arrays 28 using a motion player 30 to play the motion file. A motion file is a series of transformations that represent the motion path of a three-dimensional object through simulation space. The motion player 30 interfaces with the mechanical model 26 to control when and in what direction a given motion file plays. The motion player 30 is a software product known as VisLine that provides a lightweight visual animation capability based on CAD three-dimensional models and sets of transformations called “motion files” derived from the CAD models. The order and time for the playing of a motion file is determined by executing a program written in the PLC emulator 20. The written program is a language specification supported by the PLC emulator 20, which is capable of communicating a complex sequence for the execution of motion files and is based on the execution of specified events according to specified conditions. It should be appreciated that events cause actions to occur and create signals available as conditions for executing other events. It should also be appreciated that the transformation does not change nature of object or part but moves the object or part in space.] (FIGS. 1 and 2; Specification, page 12, line 17 through page 13, line 18).

The method includes the steps of determining whether the motion of the mechanical model is acceptable. [The method further includes determining whether the motion of the mechanical model 26 is acceptable. The user 12 views the motion or behavior of the mechanical model 26 and determines whether the behavior is acceptable or desired.] (FIGS. 1 and 2; Specification, page 13, lines 19 through 22).

The method includes the steps of replicating the motion of the mechanical model with motion commands in a PLC code using the computer if the motion of the mechanical model was acceptable. [If the behavior is acceptable, the method exports all of this information to the controls system design system 17, which converts this information to PLC code to build a

manufacturing line. The user 12 replicates the motion of the mechanical model 26 previously defined in the mechanical model 26 with motion commands in PLC code using the controls system design system 17. If the user 12 determines that the behavior is not acceptable or desired, the method includes returning to the mechanical tool design system 16 to change the mechanical model 26 as illustrated in FIG. 3.] (FIGS. 1 through 3; Specification, page 13, line 22 through page 14, line 10).

The method includes the steps of using the accepted motion of the mechanical model to compare the behavior of the PLC code to the accepted motion by playing the PLC code with a PLC emulator. [After the sequencing logic of the transformational arrays has been replicated with PLC code, the user 12 exports the PLC code to the PLC emulator 20 to play and visualize the PLC code. The user 12 compares the behavior of the PLC code to the accepted motion of the mechanical model 26 through the sequencing of the transformational arrays 28 as if they were watching a machine or manufacturing line of a vehicle assembly plant floor. It should be appreciated that the method is carried out on the computer 14 by the user 12.] (FIGS. 1 and 2; Specification, page 14, lines 11 through 14).

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

The ground of rejection to be reviewed on appeal is whether the claimed invention of claims 1 through 19 is obvious and unpatentable under 35 U.S.C. § 103 over Kanai et al. "A Virtual Verification Environment for the Sequence Control System Using VRML and JAVA", 1999 by ASME, pgs. 1 through 8 in view of Rohrer "Automod Product Suite Tutorial", Proceeding of the 1999 Winter Simulation Conference, pgs. 220-226.

ARGUMENT

Claims Not Obvious or Unpatentable Under 35 U.S.C. § 103

As to patentability, 35 U.S.C. § 103 provides that a patent may not be obtained:

If the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Id.

The United States Supreme Court interpreted the standard for 35 U.S.C. § 103 in Graham v. John Deere, 383 U.S. 1, 148 U.S.P.Q. 459 (1966). In Graham, the Court stated that under 35 U.S.C. § 103:

The scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or non-obviousness of the subject matter is determined. 148 U.S.P.Q. at 467.

Using the standard set forth in Graham, the scope and content of the prior art relied upon by the Examiner will be determined.

As to the primary reference applied by the Examiner, Kanai et al. discloses a virtual verification environment for a sequence control system using VRML and JAVA. Sequence control is one of the key components of realizing the various kind of current automated factory equipment. Building the system components in the equipment and programming the control code in a PLC are two major activities in developing the sequence control system. VRML2.0 was originally designed as the standard specification language of 3-D geometry and

its dynamic behaviors. It can be executed on any platform and any operating system. There are many VRML viewers and authoring tools commercially available, and they are much less expensive than the commercial discrete-event simulators. The specification of VRML includes the behavioral mechanism of the scene based on the event cascade. The execution engine is also installed in all VRML viewers. The language format is specified as the ISO/IEC international standard. The language can be executed in any operating system, and can be easily imported through networks. Generally, the sequence control system can be modeled as a set of finite state machines. State variables, input variables, and output variables of each component model can be defined as fields in the prototype node of the VRML. The state transitions of each model can be also described in the script nodes by combining the external JAVA code with the VRML. 3-D geometry of the components, their motion behaviors corresponding to the state transition of the component can be easily defined by adding the several standard nodes of VRML in the code. Inexpensive VRML viewer can be used for the visual verification of the co-simulation. As a result of co-simulation, the model of components in the VRML viewer (Community-Place Browser) can be dynamically moved according to the control code in the PLC.

As to the secondary reference applied by the Examiner, Rohrer discloses an automod product suite tutorial. Under user interaction, Automod provides advanced debugging and trace facilities. A model can be single-stepped at any time during the animation. Vehicle states are tracked during the entire model run, and reports are generated automatically.

Claims 1 through 10

In contradistinction, claim 1 claims the present invention as a method of emulating machine tool behavior for a programmable logic controller logical verification system

for manufacturing a motor vehicle. The method includes the steps of constructing a mechanical model using a computer, generating transformational arrays for the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer, viewing motion of the mechanical model in a motion viewer based on the transformation arrays using the computer, and determining whether the motion of the mechanical model is acceptable. The method also includes the steps of replicating the motion of the mechanical model by generating a PLC code for the motion of the mechanical model using the computer if the motion of the mechanical model was acceptable and using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator.

As to the differences between the prior art and the claims at issue, Kanai et al. merely discloses a virtual verification environment for a sequence control system using VRML and JAVA in which the sequence control system can be modeled as a set of finite state machines and state variables, input variables, and output variables. Kanai et al. lacks the steps of generating transformational arrays for a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using a computer and viewing motion of the mechanical model in a motion viewer based on the transformation arrays using the computer to determine whether the motion of the mechanical model is acceptable. In Kanai et al., during, co-simulation, there is a series of snapshots on the VRML viewer, however, there are no transformational arrays for the mechanical model that are generated by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time prior to the co-simulation on the VRML viewer to determine if the motion of the mechanical model is acceptable. The snapshots and transformational arrays are not analogous and the Examiner has

misinterpreted the Kanai et al. reference.

Kanai et al. also lacks the steps of using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator. In Kanai et al., there is no PLC emulator to play the PLC code such that the user can observe the motion of the mechanical model using the actual PLC code as if they were watching a machine or manufacturing line of a vehicle assembly plant floor as claimed by Applicants.

Rohrer merely discloses an automod product suite tutorial in which vehicle states are tracked during the entire model run and reports are generated automatically. Rohrer lacks generating transformational arrays for a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using a computer and viewing motion of the mechanical model in a motion viewer based on the transformation arrays using the computer. Rohrer also lacks using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator.

As to the level of ordinary skill in the pertinent art, in Kanai et al., during, co-simulation, there is a series of snapshots on the VRML viewer, however, there are no transformational arrays for the mechanical model that are generated by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time prior to the co-simulation on the VRML viewer to determine if the motion of the mechanical model is acceptable. In Rohrer, vehicle states are tracked during the entire model run and do not allow for transformational arrays for the mechanical model that are generated. Also, neither reference teaches generating CAD transformational arrays. Further, neither reference allows for a PLC emulator to play the PLC code such that the user can observe the motion of the

mechanical model using the actual PLC code as if they were watching a machine or manufacturing line of a vehicle assembly plant floor. However, there is absolutely no teaching of a level of skill in the programmable logic controller art that a method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle includes the steps of generating transformational arrays for a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using a computer, viewing motion of the mechanical model in a motion viewer based on the transformation arrays using the computer, and using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator. The Examiner may not, because he doubts that the invention is patentable, resort to speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in the factual basis. See In re Warner, 379 F. 2d 1011, 154 U.S.P.Q. 173 (CCPA 1967).

The present invention sets forth a unique and non-obvious combination of a method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle that allows a controls engineer to compare the behavior of the PLC code to accepted motion of a CAD model as part of PLC logical verification system and uses transformational arrays that allows a different software technology to do the rendering, one that requires much less computer resource per unit of machine, and allows a controls engineer to examine the visual behavior of an entire manufacturing line, thereby verifying some of the more difficult controls problems such as inter-workcell behavior through observation of the visual operation of multiple concurrent workcells. In this instance, the Examiner has adduced no factual basis to support his/her position that it would have been obvious to one of ordinary skill in the art to combine the teachings of Rohrer with Kanai et al.

because both references are concerned with manufacturing and material handling and it would have been convenient to track the vehicle (i.e. mechanical model) movement during the entire run, as taught by Rohrer, to the system of Kanai et al. to observe motion sequence and signal change and verify the effectiveness of the proposed modeling and implementation method. The Examiner's stated conclusion of obviousness is based on speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in the factual basis.

The references, if combinable, fail to teach or suggest the combination of a method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle including the steps of constructing a mechanical model using a computer, generating transformational arrays for a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer, viewing motion of the mechanical model in a motion viewer based on the transformation arrays using the computer, determining whether the motion of the mechanical model is acceptable, replicating the motion of the mechanical model by generating a PLC code for the motion of the mechanical model using the computer if the motion of the mechanical model was acceptable, and using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator as claimed by Applicants. Thus, the Examiner has failed to establish a case of prima facie obviousness.

Against this background, it is submitted that the present invention of claims 1 through 10 is not obvious in view of a proposed combination of Kanai et al. and Rohrer. The references cannot be combined to render obvious the combination of the method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle of claims 1 through 10. Therefore, it is respectfully submitted

that claims 1 through 10 are not obvious and are allowable over the rejection under 35 U.S.C. § 103.

Claims 11 through 19

As to claim 11, claim 11 claims the present invention as a method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle. The method includes the steps of constructing a mechanical model using a computer and generating CAD transformational arrays for motion in the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer. The method also includes the steps of constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer and viewing the motion of the motion file in a motion viewer using the computer. The method includes the steps of determining whether the motion of the mechanical model is acceptable and replicating the motion of the mechanical model with motion commands in a PLC code using the computer if the motion of the mechanical model was acceptable. The method further includes the steps of using the accepted motion of the mechanical model to compare the behavior of the PLC code to the accepted motion by playing the PLC code with a PLC emulator.

As to the differences between the prior art and the claims at issue, Kanai et al. merely discloses a virtual verification environment for a sequence control system using VRML and JAVA in which the sequence control system can be modeled as a set of finite state machines and state variables, input variables, and output variables. Kanai et al. lacks the steps of constructing a mechanical model using a computer, generating CAD transformational arrays for

motion in the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer, constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer, and viewing the motion of the motion file in a motion viewer using the computer to determine whether the motion of the mechanical model is acceptable. In Kanai et al., during, co-simulation, there is a series of snapshots on the VRML viewer, however, there are no CAD transformational arrays for the mechanical model that are generated by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time prior to the co-simulation on the VRML viewer to determine if the motion of the mechanical model is acceptable. The snapshots and transformational arrays are not analogous and the Examiner has misinterpreted the Kanai et al. reference.

Kanai et al. also lacks the steps of using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator. In Kanai et al., there is no PLC emulator to play the PLC code such that the user can observe the motion of the mechanical model using the actual PLC code as if they were watching a machine or manufacturing line of a vehicle assembly plant floor as claimed by Applicants.

Rohrer merely discloses an automod product suite tutorial in which vehicle states are tracked during the entire model run and reports are generated automatically. Rohrer lacks generating CAD transformational arrays for a mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using a computer and viewing motion of the mechanical model in a motion viewer based on the transformation arrays using the computer. Rohrer also lacks using the accepted motion of

the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator..

There is absolutely no teaching of a level of skill in the programmable logic controller art that a method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle includes the steps of constructing a mechanical model using a computer, generating CAD transformational arrays for motion in the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer, constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer, and viewing the motion of the motion file in a motion viewer using the computer to determine whether the motion of the mechanical model is acceptable, and using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator. The Examiner may not, because he doubts that the invention is patentable, resort to speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in the factual basis. See In re Warner, 379 F. 2d 1011, 154 U.S.P.Q. 173 (C.C.P.A. 1967).

Even if the references could be combined, they do not teach a level of skill in the art of programmable logic controllers of constructing a mechanical model using a computer, generating CAD transformational arrays for motion in the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer, constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer, and viewing the motion of the motion file in a motion viewer using the computer to determine whether the motion of the mechanical model is acceptable. Applicants are not attacking the references individually, but are

clearly pointing out that each reference is deficient and, if combined (although Applicants maintain that they are not combinable), the combination is deficient.

The present invention sets forth a unique and non-obvious combination of a method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle that allows a controls engineer to compare the behavior of the PLC code to accepted motion of a CAD model as part of PLC logical verification system and uses transformational arrays that allows a different software technology to do the rendering, one that requires much less computer resource per unit of machine, and allows a controls engineer to examine the visual behavior of an entire manufacturing line, thereby verifying some of the more difficult controls problems such as inter-workcell behavior through observation of the visual operation of multiple concurrent workcells. In this instance, the Examiner has adduced no factual basis to support his/her position that it would have been obvious to one of ordinary skill in the art to combine the teachings of Rohrer with Kanai et al. because both references are concerned with manufacturing and material handling and it would have been convenient to track the vehicle (i.e. mechanical model) movement during the entire run, as taught by Rohrer, to the system of Kanai et al. to observe motion sequence and signal change and verify the effectiveness of the proposed modeling and implementation method. The Examiner's stated conclusion of obviousness is based on speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in the factual basis.

The references, if combinable, fail to teach or suggest the combination of a method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle including the steps of constructing a mechanical model using a computer, generating CAD transformational arrays for motion in the mechanical model by incrementally recording one position of each piece of geometry of the

mechanical model moved through space over a period of time using the computer, constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer, viewing the motion of the motion file in a motion viewer using the computer, determining whether the motion of the mechanical model is acceptable, replicating the motion of the mechanical model with motion commands in a PLC code using the computer if the motion of the mechanical model was acceptable, and using the accepted motion of the mechanical model to compare the behavior of the PLC code to the accepted motion by playing the PLC code with a PLC emulator as claimed by Applicants.

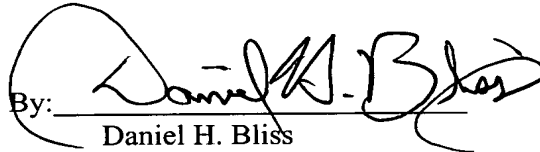
Obviousness under § 103 is a legal conclusion based on factual evidence (In re Fine, 837 F.2d 1071, 1073, 5 U.S.P.Q.2d 1596, 1598 (Fed. Cir. 1988), and the subjective opinion of the Examiner as to what is or is not obvious, without evidence in support thereof, does not suffice. Since the Examiner has not provided a sufficient factual basis which is supportive of his position (see In re Warner, 379 F.2d 1011, 1017, 154 U.S.P.Q. 173, 178 (C.C.P.A. 1967), cert. Denied, 389 U.S. 1057 (1968)), the rejection of claims 11 through 19 is improper.

Against this background, it is submitted that the present invention of claims 11 through 19 is not obvious in view of a proposed combination of Kanai et al. and Rohrer. The references cannot be combined to render obvious the combination of the method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle of claims 11 through 19. Therefore, it is respectfully submitted that claims 11 through 19 are not obvious and are allowable over the rejection under 35 U.S.C. § 103.

CONCLUSION

In conclusion, it is respectfully submitted that the rejection of claims 1 through 19 is improper and should be reversed.

Respectfully submitted,

By: 
Daniel H. Bliss
Registration No. 32,398

BLISS McGLYNN, P.C.
2075 West Big Beaver Road
Suite 600
Troy, Michigan 48084-3443
(248) 649-6090

Dated: July 16, 2007

Attorney Docket No.: 0693.00258
Ford Disclosure No.: 200-0667

CLAIMS APPENDIX

The claims on appeal are as follows:

1. A method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle, said method comprising the steps of:

constructing a mechanical model using a computer;

generating transformational arrays for the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer;

viewing motion of the mechanical model in a motion viewer based on the transformational arrays using the computer;

determining whether the motion of the mechanical model is acceptable;

replicating the motion of the mechanical model by generating a PLC code for the motion of the mechanical model using the computer if the motion of the mechanical model was acceptable; and

using the accepted motion of the mechanical model to compare the behavior of the PLC code relative to the accepted motion by playing the PLC code with a PLC emulator.

2. A method as set forth in claim 1 wherein said step of constructing comprises using a mechanical tool design system to construct the mechanical model.

3. A method as set forth in claim 2 including the step of constructing an electromechanical model.

4. A method as set forth in claim 3 wherein said step of constructing the mechanical model includes binding the electromechanical model to the mechanical model.

5. A method as set forth in claim 4 wherein said step of constructing the electromechanical model comprises using a PLC logical verification system to construct the electromechanical model.

6. A method as set forth in claim 1 wherein said step of generating transformational arrays comprises generating the transformational arrays based on computer aided design (CAD) geometries of the mechanical model.

7. A method as set forth in claim 6 including the step of exporting the mechanical model to a control system design system.

8. A method as set forth in claim 7 including the step of constructing a motion file based on the mechanical model and transformational arrays.

9. A method as set forth in claim 8 wherein said step of displaying further comprises playing the motion file by a motion player.

10. A method as set forth in claim 8 including the step of returning to the mechanical tool design system if the motion of the mechanical model is not acceptable.

11. A method of emulating machine tool behavior for a programmable logic controller logical verification system for manufacturing a motor vehicle, said method comprising the steps of:

constructing a mechanical model using a computer;

generating CAD transformational arrays for motion in the mechanical model by incrementally recording one position of each piece of geometry of the mechanical model moved through space over a period of time using the computer;

constructing a motion file based on the mechanical model and the CAD transformational arrays using the computer;

viewing the motion of the motion file in a motion viewer using the computer;

determining whether the motion of the mechanical model is acceptable;

replicating the motion of the mechanical model with motion commands in a PLC code using the computer if the motion of the mechanical model was acceptable; and

using the accepted motion of the mechanical model to compare the behavior of the PLC code to the accepted motion by playing the PLC code with a PLC emulator.

12. A method as set forth in claim 11 wherein said step of constructing comprises using a mechanical tool design system to construct the mechanical model.

13. A method as set forth in claim 12 including the step of constructing an electromechanical model.

14. A method as set forth in claim 13 wherein said step of constructing the mechanical model includes binding the electromechanical model to the mechanical model.

15. A method as set forth in claim 14 wherein said step of constructing the electromechanical model comprises using a control system design system to construct the electromechanical model.

16. A method as set forth in claim 11 wherein said step of generating comprises generating CAD transformational arrays based on computer aided design (CAD) geometries of the mechanical model.

17. A method as set forth in claim 11 including the step of exporting the mechanical model to the PLC emulator.

18. A method as set forth in claim 11 wherein said step of displaying further comprises playing the motion file by a motion player.

19. A method as set forth in claim 11 including the step of returning to the mechanical tool design system if the motion of the mechanical model is not acceptable.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None